Grimes' Fairy Tales: A 1960s Story Generator

James Ryan

Expressive Intelligence Studio
University of California, Santa Cruz
jor@soe.ucsc.edu

Abstract. We provide the first extensive account of an unknown story generator that was developed by linguist Joseph E. Grimes in the early 1960s. A pioneering system, it was the first to take a grammar-based approach and the first to operationalize Propp's famous model. This is the opening paper in a series that will aim to reformulate the prevailing history of story generation in light of new findings we have made pertaining to several forgotten early projects. Our study here has been made possible by personal communication with the system's creator, Grimes, and excavation of three obscure contemporaneous sources. While the accepted knowledge in our field is that the earliest story generator was Sheldon Klein's automatic novel writer, first reported in 1971, we show that Grimes's system and two others preceded it. In doing this, we reveal a new earliest known system. With this paper, and follow-ups to it that are in progress, we aim to provide a new account of the area of story generation that lends our community insight as to where it came from and where it should go next. We hope others will join us in this mission.

Keywords: story generation · history of the field · computational narrative

1 Introduction

We are digging up the dead: systems, documents, and other intellectual contributions from the early period of story generation that our community has, in the intervening decades, forgotten. Our project is to dismantle the prevailing history of this area, which we now know is inaccurate in several ways, to furnish a new account that tells us more about where we came from and where we should go next.[1] To carry out this project, we are scouring archives, translating documents, and engaging in personal correspondences with the scholars and system builders who were active in this period. Our discoveries include three systems that precede what is widely believed to be the earliest story generator; we are poised in a follow-up paper to nearly triple the number of known systems from before 1990. In many cases, these exhumed systems anticipate approaches that, decades later,

[1] This larger project is being conducted in collaboration with Michael Mateas and Noah Wardrip-Fruin, so I use plural pronouns in this paper.

appeared to be novel. Our scholarly process follows Wardrip-Fruin's method for *digital media archaeology* [48]: we are as interested in intellectual contexts as technical ones, and we treat systems holistically, with special attention to their underlying processes. In addition to unearthing forgotten story generators, we are also excavating historically important documents that have not been discussed or cited in decades (or ever, in some cases). Here, we present some of our findings for the first time by providing an extensive account of an essentially unknown pioneering system from the early 1960s.

But why should we care about old, forgotten work? If we view story generation as a vast design space, we can think of each implemented system as an exploratory vessel that ventures into a previously uncharted sector. If these exploratory missions are successful, they signal directions that future systems may move further into to find greater success. When success is not had, the failed projects tell us which areas to avoid. In this way, we learn about spaces that incrementalist research may push further into, dead sectors that we should not return to, and all the other still uncharted areas that we do not know much about at all. Thus, both good and bad systems generate new knowledge that is useful to contemporary and future practitioners. But when we forget about past systems—novel explorations in design space—we lose the knowledge that was generated by those systems: we forget what has been explored and what has not, and which areas are worth exploring further. As we discuss below, more than fifty years ago, the system we profile here anticipated, and then abandoned, an approach to story generation that is currently in vogue.

Beyond these fundamental practical reasons lies one of principle: as a field and as a community, we owe it to ourselves—and our forebears and our successors—to record an accurate historical record. How would you like your work to be forgotten? We, moreover, owe it to ourselves to maintain a record that encompasses not just a series of names and dates, not just a series of system architectures, but also the intellectual through lines that trace our history. Story generation is an applied technical area, but all human endeavor, especially in the area of research, has intellectual underpinnings and emerges out of intellectual contexts. Even in technical areas, there is a history of ideas that undergirds the evolution of systems over time. Returning to practical concerns, good ideas for systems can lead to bad implementations of them, and so we should track ideas too so that we might have another stab at carrying them out well.

In this paper, we provide the first extensive account of an essentially unknown story generator that was developed by linguist Joseph E. Grimes in the early 1960s. Grimes's pioneering system was the first to take a grammar-based approach and the first to operationalize Propp's famous model [40]. Is it the first ever story generator? The prevailing belief in our field is that Sheldon Klein's automatic novel writer, first reported in 1971 [28], is the earliest known system. In this paper, we show that at least three systems preceded Klein's. Thus, beyond presenting the first extensive account of Grimes's pioneering system, a broad contribution of this work is to proclaim a new earliest system. With this paper, and follow-ups to it that are in progress, we aim to provide a new account of the area of story generation that tells our community more about where we came from and where we should go next. We hope others will join us in this mission.

2 Grimes' Fairy Tales

Joseph E. Grimes earned his PhD from Cornell University in 1960, where soon after he became a professor of linguistics.[2] Since 1952, he had been a member of the Summer Institute of Linguistics (SIL), an organization based at the University of Oklahoma whose mission includes the documenting of unwritten human languages. Grimes's wife, Barbara F. Grimes, was also an SIL member and an accomplished linguist.[3] The two made their home in Mexico City, where they were documenting the Huichol language, which is spoken among indigenous peoples in the Sierra Madre Occidental. Though a field linguist, Joe Grimes had a curious interest in computers, having encountered machines—and the "statisticians and computer people" who operated them—at both Cornell and Oklahoma.

2.1 An Invitation

Upon returning to his home in Mexico City in 1960, Grimes learned of an invitation for social scientists to make use of the IBM 650 computer at the Universidad Nacional Autónoma de México there. His interest was piqued, and he took up the offer. Because he was assisting indigenous peoples in the area, the director of the university's computer lab assigned an assistant to teach Grimes how to program the machine, which used drum memory ("I had to specify where on the drum the next instruction would be") and punch cards. "My first program was a concordance maker to show the context of every morpheme in text collections of Mexican Indian languages," Grimes told us. Concordance studies typified the early history of humanities computing,[4] and the method was popular among linguists and folklorists of that time [32,12]. It is Grimes's "fifth or sixth" computer program, however, that is of special intrigue to us.

2.2 An Idea

"There was a Russian scholar, I think the name was Vladimir Propp, who had laid out an interesting hypothesis about how folk tales are put together," Grimes recalled recently. Propp's *Morphology of the Folktale*, first published in Russian in 1928, was translated into English for the first time in 1958, by Laurence Scott [13]. While this preceded the more seminal 1968 translation,[5] the work

[2] Unless otherwise noted, information and quotes provided in this section originate from personal communications with Joe Grimes (email correspondences dated June 1, 2017; June 19, 2017; June 27, 2017; and August 19, 2017).

[3] For almost thirty years, she was editor of *Ethnologue*, the preeminent catalogue of human languages.

[4] Father Busa's exhaustive indexing of all the words in the works of Thomas Aquinas [9], begun in 1949, is considered the birth of the digital humanities [25].

[5] And George Lakoff's influential 1964 reformulation using recent developments from Chomskyan linguistics [31].

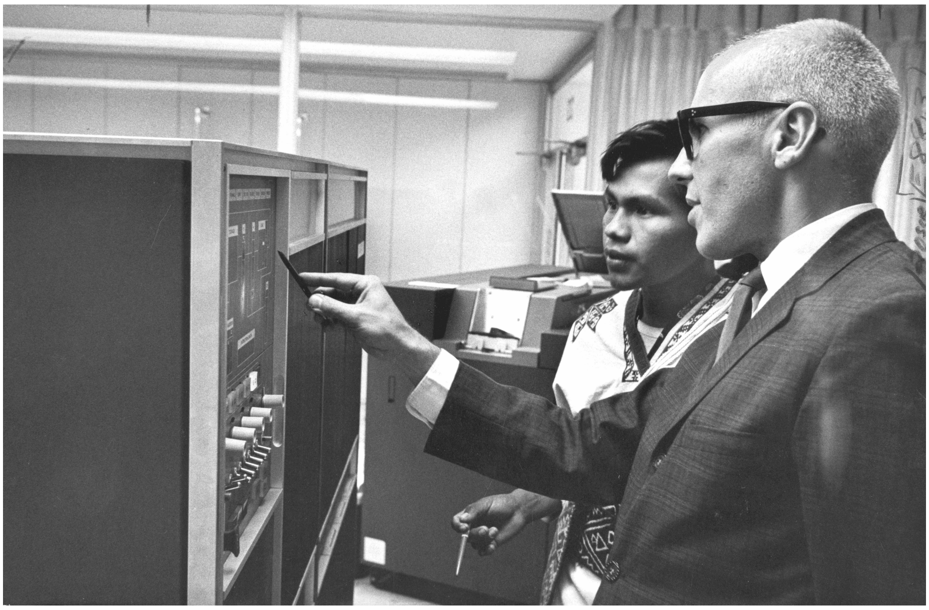

Fig. 1. In the early 1960s, linguist Joseph E. Grimes programmed an IBM computer, at the Universidad Nacional Autónoma de México, to generate stories. His system used Vladimir Propp's narrative elements and sequences and told stories in natural language prose, using simple English or Spanish. (Courtesy IBM Corporate Archives)

had an immediate impact among western scholars in a variety of disciplines, notably influencing the legendary structuralist Lévi-Strauss [13,34]. Around this time, Grimes also encountered work on narrative structure by the sociolinguist William Labov: "[Labov was] looking at the structure of nearly-got-killed stories, which he observed people would tell time after time and settle into a pattern not very different from what other people used for similar stories." With these ideas converging on him, and with his knowledge of computer programming increasing, Grimes had an idea.

2.3 A Computer Storyteller

In the summer of 1960 or 1961, Joe Grimes began programming an IBM 650 computer to tell stories. As noted above, for a decade or so linguists and folklorists had been producing machine concordances of stories, but the notion of encoding stories at a higher-level representation (*i.e.*, higher than an index of lexical items) was just beginning to emerge [10]. The notion of generating stories with a computer was especially obscure; Klein's automatic novel writer would not appear for a decade [28].[6] "I hadn't heard of anybody else trying to do what I was trying to do," Grimes told us recently.

[6] A notable exception here is SAGA II, a 1960 system that we discuss in Section 3.

```
A LION HAS BEEN IN TROUBLE FOR A LONG TIME. A DOG STEALS SOMETHING THAT
BELONGS TO THE LION. THE HERO, LION, KILLS THE VILLAIN, DOG, WITHOUT A
FIGHT. THE HERO, LION, THUS IS ABLE TO GET HIS POSSESSION BACK.
```

Fig. 2. One of the first computer-generated stories. This example tale produced by Grimes's system appeared in the IBM *Business Machines* article about the project [1]; it appears to be the only system output that survives today. The intelligent use of referring expressions and of the discourse marker 'thus' position this system at the cutting edge of natural language generation in the early 1960s.

Somebody at IBM Mexico eventually caught wind of the project and saw an opportunity: a Huichol speaker was visiting Grimes in the city, but would be leaving soon, and a Life Magazine photographer—the distinguished Cornell Capa—was available. A meeting was arranged at IBM headquarters, where Capa took photos (see Fig. 1 and Fig. 3) and another individual interviewed Grimes and his Huichol associate, who had never seen a computer before. The result of all this was a written account of Grimes's project that appeared in the May 1963 issue of *Business Machines*, IBM's employee journal. Its author wrote this about the system's processes [1, p. 11]:

> Dr. Grimes uses a basic fairy tale pattern and then programs the computer using a Monte Carlo approach so it will print out stories in a completely random manner. This involves selecting at random the path from episode to episode within the story, the action by which each episode is accomplished, and the characters which will assume the various roles. The computer can produce about 10^{20} plausible variations on a single fairy tale theme, according to Dr. Grimes.

We asked Grimes what the generated outputs looked like, and his recollection was that the stories were rendered in "simple English or Spanish." An example story is included in the IBM article (see Fig. 2), but unfortunately it appears to be the only output of the system that survives today. Grimes had a remarkable use case for his synthetic tales, as the article explains [1, p. 11]:

> "These stories are an experimental tool," says Dr. Grimes. "I present the simulated stories to a native user of a language [...] and observe his reactions to them, looking especially for places where he bogs down in trying to follow the plot. Such observations on a number of stories, and a number of different native speakers, lead to a picture of the linguistic process at work. From these observations it is possible to proceed to a hypothesis about the underlying linguistic system."

The usage of generated stories as physical field instruments is peculiar, and intriguingly anticipates current projects that are exploring the use of generated

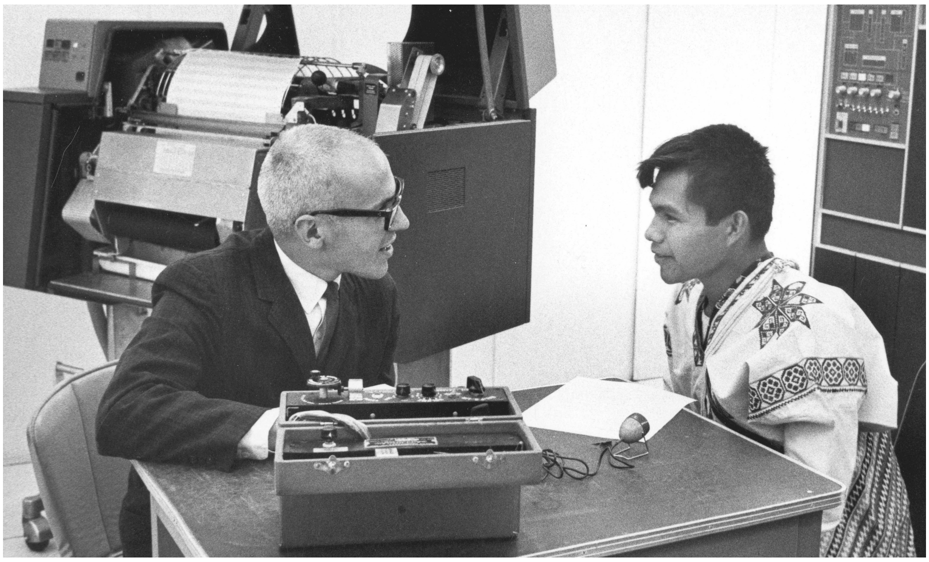

Fig. 3. Remarkably, Grimes used his generated tales as instruments in the linguistics field task of documenting the Huichol language: "I present the simulated stories to a native user of a language such as Senor Diaz [pictured] and observe his reactions to them, looking especially for places where he bogs down in trying to follow the plot. Such observations on a number of stories, and a number of different native speakers, lead to a picture of the linguistic process at work." (Courtesy IBM Corporate Archives)

stories for practical applications [41,24].[7] In the mid-1970s, Grimes would further pioneer the incorporation of computers in linguistics fieldwork by innovating, with a student, the use of portable computers in the field [46].

The system itself was produced multiple times, first on the IBM 650 and later on an IBM 1401 at the Universidad Nacional Autónoma de México, and in between on an IBM 1401 at the University of Oklahoma. Grimes programmed the 650 version in machine language and probably used Fortran for the 1401 versions. He recalls that it took a few minutes to generate a single tale, with considerable speed-up on the faster 1401.

It initially appeared to us that Grimes had not himself reported the system in any publication (and this was his recollection as well), but after scouring for some time we discovered two original sources. A 1965 volume called "The Use of Computers in Anthropology" includes an appendix with brief notes from researchers about ongoing projects, to which Grimes contributed a short write-up called "Linguistic and anthropological projects using the computer" [22]. In

[7] There are also interesting connections to one of the all-time major story generators, MEXICA [39]: both were developed in Mexico City, and while Grimes's system aided his field study of the Huichol people, MEXICA's generated stories are about the Mexica people, who are also indigenous to modern-day Mexico.

it, he notes a series of projects that had utilized the University of Oklahoma's IBM 1401 computer (emphasis ours) [22, p. 516]:

> the calculation of the Grimes-Agard index of linguistic divergence from sets of phonological correspondences, automatic positional analysis of affix systems, *Monte Carlo simulation of folk tale plots*, Program Evaluation and Review Technique analysis of field work projects, preparation and updating of bibliographies, and research into a string manipulation approach to mechanical translation

This single phrase was Grimes's only reporting of the system in an English-language publication. The other original source, which provides a more extensive description of the system, appeared as a section in a 1965 paper published in a Spanish-language linguistics journal [21]. We arranged for this section, titled 'La Simulación,' to be translated into English [23].[8] In the text, Grimes cites the specific influence of Propp and motivates the project [23, p. 1]:

> the fairy tale is a globally diffused literary genre with a very simple architecture. Several prior investigations over this topic have limited themselves to cataloguing the elements of the tale; but Propp suggests that, in addition, the global structure of the tale should be discussed, which manifests itself in each one, and which is what allows the development of new tales made according to the same global structure. I have simulated the structure that Propp defines in a computer, by way of a process that selects elements at random and orders them in the due sequence, resulting in elements for tales that have never been told, but are recognized as tales nonetheless.

He also references the more practical use of the generated tales, with additional implications for testing Propp's theory: "If the result of this process is a series of sentences or texts that the speakers of the language judge acceptable, the underlying description must be good" [23, p. 2]. Grimes recently confirmed this notion to us: "In a way it was a test of Propp's hypothesis, that certain story elements could be recognized in any story."

As for how exactly the system worked, Grimes does not have a strong recollection, but the contemporaneous sources that we have discovered provide some clues. In [23, p. 1], we find this his system operationalizes Propp's model by "a process that selects elements at random and orders them in the due sequence." This would suggest that a subset of Propp's 31 functions [40] are randomly selected and then ordered properly (Propp's list of functions is partially ordered). The example story shown in Fig. 2 initially appears to include four narrative beats, and the latter three coincide nicely with specific Proppian functions: respectively, *villainy*, *victory*, and *liquidation*. Proppian tales also include expression of an initial situation, but the opening sentence "A lion has been in trouble for a long time" corresponds better to one of the initial functions than to a

[8] Quotes here have been translated into English by Rogelio E. Cardona-Rivera.

generic initial situation. In the IBM article [1], Grimes notes that his system can generate 10^{20} unique tales, but choosing four randomly from a set of 31 would yield 31^4 combinations, a much smaller number.[9] While the example story in Fig. 2 only includes the *hero* and *villain* Proppian archetypes, there are five more. Additionally, the IBM article notes that "the action by which each episode is accomplished" is also randomly selected [1, p. 11], so each episode must have had multiple candidate actions. Still, these additional combinatorics probably do not get us to 10^{20} possibilities. Given this, we surmise that the system could generate tales with more than four elements.

A remaining issue is how the system rendered the selected elements in natural language. Our best guess is that each action was associated with a realization template, with gaps that corresponded to character archetypes being filled in with character references. Two subtle features of the generated story shown in Fig. 2 merit more explanation, however. First, the prose makes intelligent use of referring expressions: for example, 'a lion' is used the first time the character is introduced, and thereafter 'the lion' or 'the hero, lion.' Second, the discourse marker 'thus' is used to express a discourse relation between the final segment and the earlier ones. These are striking features for the time. Below, we argue that this project may have represented the state of the art of natural language generation in the early 1960s.

2.4 Aftermath

The 1963 IBM article noted this about the future of the project [1, p. 11]:

> Dr. Grimes has laid out plans for a volume of computer-produced folk tales that linguists anywhere can use as an experimental tool in their studies of language. Dr. Grimes' colleagues have suggested that an appropriate title for the book might be *Grimes' Fairy Tales*.

This book never appeared for the same reason that Grimes did not write more about the project—the generated outputs were not good: "The thing I never put my finger on was that my computer's stories had Propp's elements and sequences, but they were all boring." By the mid-1960s, Grimes had abandoned the project: "I had too much going on successfully in other areas, so I decided to leave this one for somebody else." Others did come along.

In 1968, Robert I. Binnick was a research assistant in Victor Yngve's lab at the University of Chicago who was working alone on a curious side project.[10] Inspired by his mentor's pioneering work in natural language generation [50], as well as George Lakoff's recent revision to Propp's model [31], Binnick was developing a Proppian story generator of his own. The system, as he recalls, was a "COMIT program instantiating a (probably) context-free phrase-structure

[9] And ordering constraints would further reduce the size of this space.

[10] Unless otherwise noted, information and quotes about this project originate from personal communications with Bob Binnick (email correspondences dated June 23, 2017, and August 9, 2017).

grammar." It could generate "plot outlines or at best a series of statements of events," but the results were "nothing like a real story that a story-teller would tell." Like Grimes, Binnick did not think his results merited publication, and the system was only reported offhandedly: in a sentence-and-a-half aside of a 1969 paper, he noted that his generated stories were "partly abominable and partly amusing" [7, p. 27]. Until now, Binnick's system was also forgotten.

A few years later, in 1971, Sheldon Klein introduced his system for generating murder mysteries [28], the first widely reported effort in story generation. Klein and Grimes were friends who were "tuned to the same frequencies" and ran into each other at (computational) linguistics meetings, but they never worked together. Grimes recalls them discussing stories, but he was not aware of Klein's system and assumes Klein was not aware of his ("I didn't think of it as anything to brag about"). Klein never mentioned Grimes in his work, instead citing Roald Dahl's short story "The Great Automatic Grammatizator" as his primary influence [28, p. 2].

Grimes's system would likely not be known today had the notable folklorist Alan Dundes not mentioned it in his 1965 paper "On Computers and Folk Tales" [12, p. 188].[11] The paper is about computer *encoding* of stories, but in a footnote Dundes says this about Grimes's project before pointing to the IBM article:

> It should be realized that there are other possible uses for computers in the study of folk tales. For example, computers can be used to generate folk tales. Linguist Joseph E. Grimes has succeeded in programming the essence of Vladimir Propp's morphological pattern of folk tales and he has been able to generate tales. Grimes plans to use these artifical [sic] tales for "planting" experiments in "the field."

Before abandoning the project, Grimes demonstrated his system in a seminar at Kansas State University that was held in December 1965 [2,3,4,5,6]. In the week leading up to it, a local newspaper described the event in this way [6]: "Grimes will run a simulation of folk-tale plots by random process, enabling a 1620 computer to compose folk tales, one after the other." Intriguingly, reports also stated that the demonstration would be filmed for later use as training materials [4,6]. We contacted the Kansas State University archives to ask whether any footage survives today,[12] but none could be found—an archivist there suspected that the film would have been stored in a building that was burned down in 1968. It is still possible that other materials related to Grimes's project exist today—code listings, punch cards, or printouts, perhaps—but everything that has turned up so far has been reported in this paper.

3 The First Story Generator?

Essentially all publications in the area of story generation cite Klein's automatic novel writer [28] as the earliest known system (*e.g.*, [19]), and a few make the

[11] Grimes's own brief accounts in English and Spanish have never been cited.
[12] Email correspondence dated July 13, 2017.

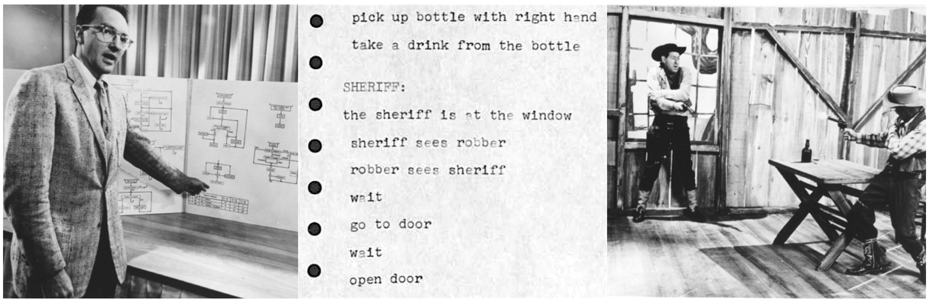

Fig. 4. The earliest known story generator is a 1960 MIT system called SAGA II, which generated scenes from television screenplays in the western genre. CBS showcased the system on a program called "The Thinking Machine," which featured discussion of its architecture (left) and live-action renditions of three generated scenes (right); actual system output is also shown here (middle). (Courtesy Computer History Museum)

worse mistake of attributing this to TALE-SPIN, which was introduced even later, in 1975 [36] (*e.g.*, [14]). That being said, a handful of scholars *have* become aware of Grimes's project, through the footnote in Dundes's paper [12, p. 188].[13] Unfortunately, every one of these scholars has misattributed the system to Dundes [45,33,15,43] or to a collaboration between Dundes and Grimes [16,17,29]. Thus, beyond providing the first extensive account of this project, this paper is also the first to correctly attribute the system to its true author, Grimes.

While Grimes's 1963 story generator predates Klein's 1971 effort,[14] it is not the earliest known story generator. In October 1960, CBS honored the upcoming 1961 centennial of the Massachusetts Institute of Technology (MIT) by producing a television series called *Tomorrow*. One episode, called "The Thinking Machine," profiled uses at MIT of the TX-0 digital computer, including a program named SAGA II that generated screenplays in the TV western genre (see Fig. 4).[15] Mark Sample rediscovered this material in 2013 and wrote about SAGA II in the context of randomness in expressive systems [44], but until now no one has articulated its place as the earliest known story generator; this is another contribution of this paper. In the CBS program, the SAGA II architecture is discussed and three generated scenes are acted out, including a degenerate one in the style of Meehan's 'mis-spun tales' [37]. A technical memorandum between the system programmer and the project lead also survives [38], which gives some insight into the system's processes: scenes are broken into narrative beats, each of which is defined as a branching action structure that is traversed probabilistically according to the story state. The system's operation is illustrated in the CBS program using a tidy flow chart. Intriguingly, this architecture anticipates

[13] In our field, Lee's 1994 master's thesis is the earliest such work [33].
[14] And indeed Binnick's 1969 system does too.
[15] The episode is available online at `http://techtv.mit.edu/videos/10268-the-thinking-machine-1961---mit-centennial-film`; the segment of interest begins around the 32-minute mark.

Klein's murder-mystery generator, which Marie-Laure Ryan characterizes using a probabilistic flow chart in an unknown (but very good) 1987 paper [42].

If Grimes began developing his system in the summer of 1960—a possibility that he has stated to us—then perhaps his system existed in some form before SAGA II did. The safest bet, though, is to rely on the publication dates of extant materials, and in that regard SAGA II precedes his system by at least one and a half years.[16] Curiously, in the 1963 IBM article, Grimes references the rigid narrative structure of TV westerns [1, p. 11]:

> "All fairy tales contain a number of segments," says Dr. Grimes, "and there are definite constraints on the order in which these segments can be placed. In a rather cramped subtype of the fairy tale known as the TV western, for example, the gunslinger in the black hat can be done in only after he has perpetrated some kind of villainy. To arrest him for vagrancy at the outset and keep him in jail until the final commercial, would violate the esthetic sense of every third grader in the land."

We asked Grimes if he recalled ever seeing the CBS program, and specifically whether SAGA II could have been an influence on his project, but he was sure that he had not been aware of the work.

4 Conclusion

Grimes's system is incredibly important to our field. If it is not the first story generator, since SAGA II was reported first, it is still the earliest known system to generate complete stories (SAGA II only generated scenes). More specifically, it is the earliest known Proppian system, preceding even the seminal second translation of Propp into English in 1968 and Lakoff's influential 1964 reformulation of the model [31]. Now, in the present day, we are experiencing a revival in Proppian story generation (*e.g.*, [20]), which could be considered curious given that Grimes (and also Binnick) found the approach to be unworthy more than fifty years ago. More broadly, Grimes's system is the first to use *story grammars*, a major 1970s approach that until recently had been largely abandoned in light of Black and Wilensky's famous 1979 skewering [8].[17] As the present repeats the (until now) forgotten past, we wonder how the history of story generation (and narrative modeling more broadly) would be altered had Grimes reported, as early as the mid-1960s, the negative result of his Proppian system that could generate 10^{20} "boring" tales.[18]

[16] Another consideration is how 'story generation' is defined, but we will leave that discussion for a different paper.

[17] Story grammars did not go down without a fight [35], and their appeal in contemporary contexts is evidenced by Kate Compton's Tracery [11].

[18] To be clear, we are not calling for the abandonment of Proppian story generation. Rather, we mean to shed light on the origins of this approach, and in this light we find that the earliest attempts were curiously aborted. In any event, even successful entrants in this tradition should acknowledge their forebears: Grimes and Binnick.

Finally, there is another aspect of Grimes's project that merits discussion. While the generated prose of the example story shown in Fig. 2 may not impress readers today, it has features that suggest the need to reappraise the early history of natural language generation. First, let us summarize the prevailing historical account of this area. In 1961, Victor Yngve produced a seminal paper reporting a system that could produce grammatically correct nonsense sentences, *e.g.*, "A proud, little, proud and heated headlight is little and shiny" [50, p. 77]. AUTO-BEATNIK, one of the earliest computer poetry generators, used a similar technique to produce sentences of comparable quality: "All blows have glue, few toothpicks have wood" [49, p. 97]. This was the state of the art of natural language generation in 1962 [18], and the following year Sheldon Klein and Robert F. Simmons pushed the cutting edge a bit farther by maintaining dependency structure within generated sentences [27]. The first reported system to generate discourse comprising multiple sentences appeared in Klein's follow-up work of 1965 [26]. Grimes's system, we now know, did this as early as 1963. Moreover, as the generated story in Fig. 2 demonstrates, the system made intelligent use of referring expressions—'a lion' is used the first time the character is introduced, and thereafter 'the lion' or 'the hero, lion'—and also a discourse marker, 'thus'. Given these novel features, and what we know about the state of the art in the early 1960s, it would seem that Grimes's system was among the most advanced natural language generation projects of its time.

This paper is the first in a series on the forgotten early history of computer story generation. Until now, Grimes's pioneering system had been known to only a handful of scholars, each of whom misattributed it to another creator. We have provided here the first extensive account of the project, made possible by personal communication with its true creator and excavation of three obscure contemporaneous sources [1,22,23]. In addition to profiling Grimes's 1963 story generator, we have also introduced another early Proppian system—Binnick's 1969 one—and have been the first to declare that the 1960 SAGA II system [38] is now the earliest known story generator.[19] Thus, while the prevailing belief in our field has been that Klein's automatic novel writer [28] was the first story generator, we have shown that it has at least three antecedents. We stated at the beginning of this paper that our larger project is to dismantle the prevailing history of this area, which, as we have begun to show here, is inaccurate in several ways. Eventually, we will furnish a comprehensive account that integrates all of our findings. While others have conducted technical reviews of the field [19,30] and deep dives into specific systems [47], our larger project is the first attempt at an extensive intellectual history. We aim specifically for a rich account that tells us more about where we came from and where we should go next, and we hope that others will join us in this mission.

[19] Again, we note that Mark Sample rediscovered this system in 2013 [44].

5 Acknowledgments

We are deeply indebted to Joseph E. Grimes, who, over the span of two months, graciously answered numerous questions about his project. Likewise, we thank Robert I. Binnick for taking the time to respond to inquiries regarding his own pioneering system. Rogelio E. Cardona-Rivera pitched in to translate Grimes's Spanish-language account of his system—this translation proved to be a critical source. L.J. Strumpf, of the IBM Corporate Archives, furnished another major source, the *Business Machines* article. He also provided high-quality scans of the archival images included in this paper—these have not been seen since the photographs were taken in 1963. Finally, we would like to thank Cliff Hight, archivist at Kansas State University, who also provided assistance on the project.

References

1. Exploring the fascinating world of language. Business Machines XLVI(3), 10–11 (1963), courtesy IBM Corporate Archives
2. Language to be discussed. Kansas State Collegian 72(60), p. 6 (Dec 14 1965)
3. Linguist demonstrates shortcut to translating. Kansas State Collegian 72(60), p. 1 (Dec 14 1965)
4. Linguist to explain methods. Kansas State Collegian 72(53), p. 4 (Dec 7 1965)
5. Linguistics expert to give demonstration of methods. Kansas State Collegian 72(59), p. 1 (Dec 13 1965)
6. Linguists demonstration Monday at Kansas State. The Manhattan (Kansas) Mercury p. 1 (Dec 7 1965)
7. Binnick, R.I.: An application of an extended generative semantic model of language to man–machine interaction. In: Proc. Conference on Computational Linguistics. pp. 1–34 (1969)
8. Black, J.B., Wilensky, R.: An evaluation of story grammars. Cognitive Science 3(3), 213–229 (1979)
9. Busa, R.: The annals of humanities computing: The index thomisticus. Computers and the Humanities 14(2), 83–90 (1980)
10. Colby, B.N., Collier, G.A., Postal, S.K.: Comparison of themes in folktales by the general inquirer system. The Journal of American Folklore 76(302), 318–323 (1963)
11. Compton, K., Kybartas, B., Mateas, M.: Tracery: an author-focused generative text tool. In: Proc. International Conference on Interactive Digital Storytelling. pp. 154–161. Springer (2015)
12. Dundes, A.: On computers and folk tales. Western Folklore 24(3), 185–189 (1965)
13. Dundes, A.: Binary opposition in myth: The Propp/Lévi-Strauss debate in retrospect. Western Folklore 56(1), 39–50 (1997)
14. Eger, M., Potts, C.M., Barot, C., Young, R.M.: Plotter: Operationalizing the master book of all plots. In: Proc. Intelligent Narrative Technologies and Social Believability in Games. pp. 30–33 (2015)
15. Fairclough, C.: Story games and the OPIATE system. Ph.D. thesis, University of Dublin, Trinity College (2004)
16. Ferri, G.: Between procedures and computer games: Semiotics of practices as a unifying perspective. In: Proc. International Association for Semiotic Studies. pp. 291–302. Universidade da Coruña (2012)

17. Ferri, G.: Narrative structures in IDN authoring and analysis. In: Koenitz, H., Ferri, G., Haahr, M., Sezen, D., Sezen, T.İ. (eds.) Interactive Digital Narrative: History, Theory and Practice, pp. 77–90. Routledge (2015)
18. Garvin, P.L.: Computer participation in linguistic research. Language 38(4), 385–389 (1962)
19. Gervas, P.: Computational approaches to storytelling and creativity. AI Magazine 30(3), 49 (2009)
20. Gervás, P.: Propp's morphology of the folk tale as a grammar for generation. In: Proc. Computational Models of Narrative. pp. 106–122 (2013)
21. Grimes, J.E.: La computadora en las investigaciones humanísticas. Anuario de Letras. Lingüística y Filología 5, 163–174 (1965)
22. Grimes, J.E.: Linguistic and anthropological projects using the computer. The Use of Computers in Anthropology pp. 515–516 (1965)
23. Grimes, J.E., Cardona-Rivera, R.E., Ryan, J.: Translation of "La Simulación" (2017), available upon request
24. Harrison, B., Riedl, M.O.: Learning from stories: Using crowdsourced narratives to train virtual agents. In: Proc. Artificial Intelligence and Interactive Digital Entertainment. pp. 183–189 (2016)
25. Hockey, S.: The history of humanities computing. A Companion to Digital Humanities pp. 3–19 (2004)
26. Klein, S.: Automatic paraphrasing in essay format. Mechanical Translation & Compuational Linguistics 8(3-4), 68–83 (1965)
27. Klein, S., Simmons, R.F.: Syntactic dependence and the computer generation of coherent discourse. Mechanical Translation 7(2), 50–61 (1963)
28. Klein, S., et al.: A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes. University of Wisconsin Tech. Rep. TR142 (1971)
29. Koenitz, H., Ferri, G., Haahr, M., Sezen, D., Sezen, T.İ.: Introduction: The evolution of interactive digital narrative theory. In: Koenitz, H., Ferri, G., Haahr, M., Sezen, D., Sezen, T.İ. (eds.) Interactive Digital Narrative: History, Theory and Practice, pp. 69–76. Routledge (2015)
30. Kybartas, B., Bidarra, R.: A survey on story generation techniques for authoring computational narratives. Transactions on Computational Intelligence and AI in Games (2016)
31. Lakoff, G.: Structural complexity in fairy tales. The Study of Man 1, 128–150 (1972)
32. Lamb, S.M., Gould, L.: Concordances from computers (1964)
33. Lee, M.: A model of story generation. Master's thesis, University of Manchester, Manchester (1994)
34. Lévi-Strauss, C., Weightman, J., Weightman, D.: The raw and the cooked: Introduction to a science of mythology. Pimlico New York, NY (1994)
35. Mandler, J.M., Johnson, N.S.: On throwing out the baby with the bathwater: A reply to Black and Wilensky's evaluation of story grammars. Cognitive Science 4(3), 305–312 (1980)
36. Meehan, J.R.: Using planning structures to generate stories. American Journal of Computational Linguistics 33, 78–94 (1975)
37. Meehan, J.R.: The metanovel: Writing stories by computer. Ph.D. thesis, Yale University (1976)
38. Morse, H.R.: Preliminary operating notes for SAGA II. MIT Technical Memorandum 8436-M-29 (1960)

39. Pérez ý Pérez, R., Sharples, M.: Mexica: A computer model of a cognitive account of creative writing. Journal of Experimental & Theoretical Artificial Intelligence 13(2), 119–139 (2001)
40. Propp, V.: Morphology of the Folktale. University of Texas Press (2010)
41. Riedl, M.O., Harrison, B.: Using stories to teach human values to artificial agents. In: Proc. AI, Ethics, and Society. pp. 105–112 (2016)
42. Ryan, M.L.: The heuristics of automatic story generation. Poetics 16(6), 505–534 (1987)
43. Sack, W.: Une machine à raconter des histoires: Propp et les software studies. Les Temps Modernes (5), 216–243 (2013)
44. Sample, M.: An account of randomness in literary computing (2013), presented at Modern Language Association
45. Smith, J.B.: Thematic structure and complexity. Style 9, 32–54 (1975)
46. Summer Institute of Linguistics: First portable computer for linguistic fieldwork, `https://www.sil.org/history-event/first-portable-computer-linguistic-fieldwork`, accessed 22 June 2017
47. Wardrip-Fruin, N.: Expressive Processing: Digital fictions, computer games, and software studies. MIT Press (2009)
48. Wardrip-Fruin, N.: Digital media archaeology: Interpreting computational processes. Media archaeology: Approaches, applications, and implications pp. 302–322 (2011)
49. Worthy, R.M.: A new American poet speaks: The works of A.B. Horizon: A Magazine of the Arts IV(5), 96–99 (1962)
50. Yngve, V.H.: Random generation of English sentences. In: Proc. Machine Translation of Languages and Applied Language Analysis. pp. 66–80 (1961)